








Digital Electronics and Analog Photonics for Convolutional Neural Networks (DEAP-CNNs)

Viraj Bangari , Bicky A. Marquez, *Member, IEEE*, Heidi Miller , Alexander N. Tait , *Member, IEEE*, Mitchell A. Nahmias , Thomas Ferreira de Lima , *Student Member, IEEE*, Hsuan-Tung Peng , Paul R. Prucnal, *Life Fellow, IEEE*, and Bhavin J. Shastri , *Senior Member, IEEE*

Abstract—Convolutional Neural Networks (CNNs) are powerful and highly ubiquitous tools for extracting features from large datasets for applications such as computer vision and natural language processing. However, a convolution is a computationally expensive operation in digital electronics. In contrast, neuromorphic photonic systems, which have experienced a recent surge of interest over the last few years, propose higher bandwidth and energy efficiencies for neural network training and inference. Neuromorphic photonics exploits the advantages of optical electronics, including the ease of analog processing, and busing multiple signals on a single waveguide at the speed of light. Here, we propose a Digital Electronic and Analog Photonic (DEAP) CNN hardware architecture that has potential to be 2.8 to 14 times faster while using almost 25% less energy than current state-of-the-art graphical processing units (GPUs).

Index Terms—Deep learning, machine learning, neuromorphic photonics, photonic neural networks, convolutional neural network (CNN).

I. INTRODUCTION

THE success of convolutional neural networks (CNNs) for large-scale image recognition has stimulated research in developing faster and more accurate algorithms for their use. However, CNNs are computationally intensive and therefore results in long processing latency. One of the primary

bottlenecks is computing the matrix multiplication required for forward propagation. In fact, over 80% of the total processing time is spent on the convolution [1]. Therefore, techniques that improve the efficiency of even forward-only propagation are in high demand and researched extensively [2], [3].

In this work, we present a complete digital electronic and analog photonic (DEAP) architecture capable of performing highly efficient CNNs for image recognition. The competitive MNIST handwriting dataset [4] is used as a benchmark test for our DEAP CNN. At first, we train a standard two-layer CNN offline, after which network parameters are uploaded to the DEAP CNN. Our scope is limited to the forward propagation, but includes power and speed analyses of our proposed architecture.

Due to their speed and energy efficiency, photonic neural networks have been widely investigated from different approaches that can be grouped into three categories: (1) reservoir computing [5]–[8]; reconfigurable architectures based on (2) ring-resonators [9]–[12], and (3) Mach-Zehnder interferometers [13], [14]. Reservoir computing in the discrete photonic domain successfully implement neural networks for fast information processing, however the predefined random weights of their hidden layers cannot be modified [8].

An alternative approach uses silicon photonics to design fully programmable neural networks [15], using a so-called broadcast-and-weight protocol [10]–[12]. This scheme is based on wavelength-division multiplexing (WDM) and is capable of implementing reconfigurable, recurrent and feedforward neural network models, using a bank of tunable silicon microring resonators (MRRs) that recreate on-chip synaptic weights. Therefore, such a protocol allows it to emulate physical neurons. Mach-Zehnder interferometers (MZIs) have been also used to model synaptic-like connections of physical neurons [14]. Fully reconfigurable photonic architectures based on MZI arrays are introduced in the work of Bagherian *et al.* [16]. This architecture uses constructive or destructive interference effects in MZIs to implement a matrix-vector operation on photonic signals. This approach is limited to using a single wavelength of light, and being all-optical, the architecture must grapple with both amplitude and phase, with challenges of phase noise accumulation from one nonlinear stage to another.

The advantage of the broadcast-and-weight approach over the coherent approach is that it has already demonstrated fan-in, inhibition, time-resolved processing, and autaptic cascability [12]. The DEAP-CNN architecture is therefore designed

Manuscript received April 18, 2019; revised August 14, 2019; accepted August 31, 2019. Date of publication October 4, 2019; date of current version November 4, 2019. The work of V. Bangari, B. A. Marquez, and B. J. Shastri was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Queen's Research Initiation Grant (RIG). The work of B. A. Marquez was also supported by the 2019 Queens Postdoctoral Fund. (Corresponding authors: Viraj Bangari; Bhavin J. Shastri.)

V. Bangari, B. A. Marquez, and H. Miller are with the Department of Physics, Engineering Physics & Astronomy, Queen's University, Kingston, ON K7L 3N6, Canada (e-mail: viraj.bangari@queensu.ca; bama@queensu.ca; miller.heidi@queensu.ca).

A. N. Tait was with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA. He is now with the Physical Measurement Laboratory, National Institute of Standards and Technology (NIST), Boulder, CO 80305 USA (e-mail: atait@ieee.org).

M. A. Nahmias, T. Ferreira de Lima, H.-T. Peng, and P. R. Prucnal are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: mnahmias@princeton.edu; tlima@princeton.edu; hpeng@princeton.edu; prucnal@princeton.edu).

B. J. Shastri is with the Department of Physics, Engineering Physics & Astronomy, Queen's University, Kingston, ON K7L 3N6, Canada, and also with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: shastri@ieee.org).

Color versions of one or more of the figures in this article are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTQE.2019.2945540

using banks of tunable MRRs. The DEAP-CNN design is compatible with mainstream silicon photonic device platforms. Consequently, this approach leverages the advances in silicon photonics that have recently progressed to the level of sophistication required for large-scale integration. Furthermore, this proposed architecture allows the implementation of multi-layer networks to implement the deep learning framework.

Inspired by the work of Mehrabian *et al.* [17], which lays out a potential architecture for photonic CNNs, our design goes a step further by considering specific details about a complete physical implementation, as well as an example of how an algorithm for tasks such as MNIST can be mapped to photonics. In particular, our approach exploits the use of multi-dimensional kernels and inputs, and the summation of multi-channel inputs that leads us to implement on-chip general photonic convolutions. Furthermore, we break the limitations on weights being between -1 and $+1$ by introducing transimpedance amplifiers (TIAs), which is a unique feature of our proposed DEAP-accelerator, and has not been considered in related recent works [18].

This work is divided in five sections: Following this introduction, in Section (II), we describe convolutions as used in the field of signal processing. Then, we introduce silicon photonic devices to perform convolutions in photonics. Section (III) introduces a hardware inspired algorithm to perform such full photonic convolutions. In Section (IV), we utilize our previously described architecture to build a two-layer DEAP CNN for MNIST handwritten digit recognition. Finally, in Section (V), we show an energy-speed benchmark test, where we compare the performance of DEAP with the empirical dataset DeepBench [19]. Note, we have made the high level simulator and mapping tool written in Python for the DEAP architecture publicly available [20].

II. CONVOLUTIONS AND PHOTONICS

A. Convolutions Background

A convolution of two discrete domain functions f and g is defined by:

$$(f * g)[t] = \sum_{\tau=-\infty}^{\infty} f[\tau]g[t - \tau], \quad (1)$$

where $(f * g)$ represents a weighted average of the function $f[\tau]$ when it is weighting by $g[-\tau]$ shifted by t . The weighting function $g[-\tau]$ emphasizes different parts of the input function $f[\tau]$ as t changes.

In digital image processing, a similar process is followed. The convolution of an image A with a kernel F produces a convolved image O . An image is represented as a matrix of numbers with dimensionality $H \times W$, where H and W are the height and width of the image, respectively. Each element of a matrix represents the intensity of a pixel at that particular spatial location. A kernel is a matrix of real numbers with dimensionality $R \times R$. The value of a particular convolved pixel is defined by:

$$O_{i,j} = \sum_{k=1}^R \sum_{l=1}^R F_{k,l} A_{i+k,j+l}. \quad (2)$$

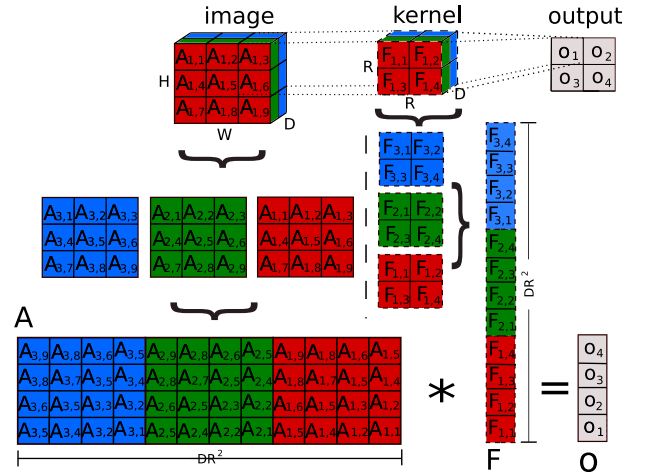


Fig. 1. Schematic illustration of a convolution. At the top of the figure, an input image is represented as a matrix of numbers with dimensionality $H \times W \times D$ where H , W , and D are the height, width, and depth of the image, respectively. Each element $A_{i,j}$ of A represents the intensity of a pixel at that particular spatial location. The kernel F is a matrix with dimensionality $R \times R \times D$, where each element $F_{i,j}$ is defined as a real number. The kernel is slid over the image by using a stride S equal to one. As the image has multiple channels (or depth) D , the same kernel is applied to each channel. Assuming $H = W$, the overall output dimensionality is $(H - R + 1)^2$. The bottom of the figure shows how a convolution operation generalized into a single matrix-matrix multiplication, where the kernel F is transformed into a vector \mathbf{F} with DR^2 elements, and the image A is transformed into a matrix \mathbf{A} of dimensionality $DR^2 \times (H - R + 1)^2$. Therefore, the output is represented by a vector with $(H - R + 1)^2$ elements.

Using matrix slicing notation, Eq. (2) can be represented as a dot product of two vectorized matrices:

$$O_{i,j} = \text{vec}(F)^T \cdot \text{vec}((A_{m,n})_{n \in [j,j+R]}^{m \in [i,i+R]}).^T. \quad (3)$$

A convolution reduces the dimensionality of the input image to $(H - R + 1) \times (W - R + 1)$, so a padding of zero values is normally applied around the edges of the input image to counteract this. A schematic illustration of a convolution in digital image processing is shown at the top of Fig. 1.

When convolutions are used to perform parallel matrix multiplications in neural networks such as CNNs, a convolution operation is defined as:

$$O_{i,j} = \text{vec}(F)^T \cdot \text{vec}((A_{m,n,k})_{k \in [1,D]}^{m \in [iS,iS+R]}_{n \in [jS,jS+R]}).^T, \quad (4)$$

where the input A has dimensionality $H \times W \times D$, kernel F has dimensionality $R \times R \times D$ and D refers to the number of channels within the input image. The additional parameter S is referred to as the “stride” of the convolution. This convolution is similar to Eq. (3), except that the outputs from each channel are summed together in the end, and that the stride parameter is always equal to 1 in image processing. The dimensionality of the output feature is:

$$\left\lceil \frac{H-R}{S} + 1 \right\rceil \times \left\lceil \frac{W-R}{S} + 1 \right\rceil \times K, \quad (5)$$

where K is the number of different kernels applied to an image, and $\lceil \cdot \rceil$ is the ceiling function. Table I contains a summary of all the convolutional parameters described so far.

TABLE I
SUMMARY OF CONVOLUTIONAL PARAMETERS

Parameter	Meaning
N	Number of input images
H	Height of input image including padding
W	Width of input image including padding
D	Number of input channels
R	Edge length of kernel
K	Number of kernels
S	Stride

One of the challenges with convolutions is that they are computationally intensive operations, taking up 86% to 94% of execution time for CNNs [1]. For heavy workloads, convolutions are typically run on graphical processing units (GPUs), as they are able to perform many mathematical operations in parallel. A GPU is a specialized hardware unit that is capable of performing a single mathematical operation on large amounts of data at once. This parallelization allow GPUs to compute matrix-matrix multiplication at speeds much higher than a CPU [21]. The convolution operation can be generalized into a single matrix-matrix multiplication [22]. This is shown at the bottom of Fig. 1, where the kernel F is transformed into a vector \mathbf{F} with dimensionality $KDR^2 \times 1$, and the image is transformed into a matrix \mathbf{A} of dimensionality $KDR^2 \times \lceil \frac{H-R}{S} + 1 \rceil \lceil \frac{W-R}{S} + 1 \rceil K$. Therefore, the output is represented by a vector with $\lceil \frac{H-R}{S} + 1 \rceil \lceil \frac{W-R}{S} + 1 \rceil K$ elements; where in this particular case $K = 1$, $S = 1$ and $H = W$.

B. Silicon Photonics Background

An emerging alternative to GPU computing is optical computing using silicon photonics for ultrafast information processing. Silicon photonics is a technology that allows for the implementation of photonic circuits by using the existing complementary-metal-oxide-semiconductor (CMOS) platform for electronics [23]. In recent years, the silicon photonic based “broadcast-and-weight” architecture has been shown to perform multiply-accumulate operations at frequencies up to five times faster than conventional electronics [24]. The broadcast-and-weight protocol employs a bank of tunable silicon MRRs that recreate on-chip synaptic weights. Therefore, the five times speed improvement of the broadcast-and-weight architecture is enabled by the modulation/switching speed of a silicon microring modulator. The state-of-the-art tuning speed of silicon microring is at 10–40 GHz due to the plasma dispersion effect [25], which is the effect that the change of free carriers density in semiconductors results in the change of both real and imaginary parts of the refractive index. On the other hand, the electronic processors have their clock rate limit at around 4–5 GHz as they reach the thermal dissipation limit. Therefore, there is a motivation to explore how photonics could be used to perform convolutions, and how it compares to GPU-based implementations.

MRRs are the building block of our approach and are used to map input and kernel values to photonics. A MRR is a circular waveguide that is coupled with either one or two waveguides.

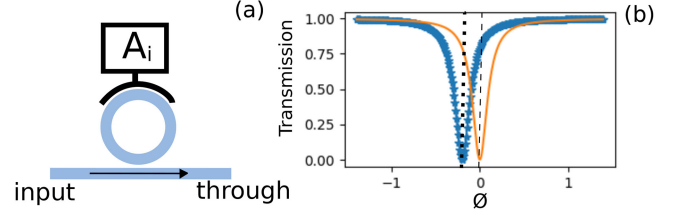


Fig. 2. (a) All-pass MRR and (b) transfer function: The orange curve represents the Lorentzian line shape described by Eq. (6), centered in the initial phase where MRR is in resonance with the incoming light. The blue triangle curve shows how such phase can be modified by heating the MRR via the application of a current through A_i .

Such silicon waveguides can be manufactured to have a width of 500 nm while having a thickness of 220 nm. These waveguides have a bend radius of 5 μm and can support transverse-electric (TE) and transverse-magnetic (TM) polarized wavelengths between 1.5 μm and 1.6 μm [23]. The single waveguide configuration is called an all-pass MRR as shown in Fig. 2(a).

The light from the waveguide is transferred into the ring via a directional coupler and then recombined. The effective index of refraction between the waveguide and the MRR and the circumference of the MRR cause the recombined wave to have a phase shift, thereby interfering with the intensity of the original light. The transfer function of the intensity of the light coming out at the through port with the light going into the input port of the all-pass resonator is described by:

$$T_n(\phi) = \frac{a^2 - 2ra \cos(\phi) + r^2}{1 - 2ra \cos(\phi) + (ar)^2}. \quad (6)$$

The parameter r is the self-coupling coefficient, and a defines the propagation loss from the ring and the directional coupler. The phase ϕ depends on the wavelength λ of the light and radius d of the MRR [26]:

$$\phi = \frac{4\pi^2 dn_{eff}}{\lambda}, \quad (7)$$

where n_{eff} is the effective index of refraction between the ring and waveguide. The value of n_{eff} can be modified to indirectly change the resonance peak. Such tuning is usually made by applying a current across embedded silicon in-ring photoconductive heater [27]. This process heats the ring, yielding a shift of the resonance peak. Fig. 2(b) shows an example of such tuning: the orange curve represents the Lorentzian line shape described by Eq. (6), centered in the initial phase of the ring resonator, indicating that the MRR is in resonance with the incoming light. The blue triangle curve shows how such phase can be modified by heating the MRR.

The phase for an all-pass resonator corresponding to a particular intensity modulation value can be computed by using Eq. (6):

$$\phi_i = \arccos \left[\frac{A_i(1 + (ar)^2) - a^2 - r^2}{2ra(1 - A_{i,j})} \right], \quad (8)$$

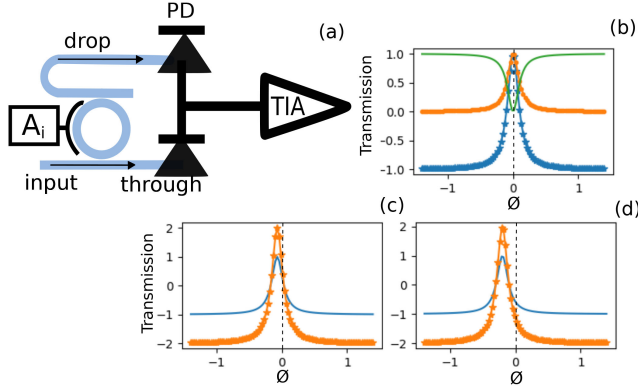


Fig. 3. (a) Add-drop configuration and O/E conversion and amplification. (b) Output of the balanced photodiode, the transfer function of $T_d - T_p$. The orange circle and green curves are drop and through ports, described by Eq. (11) and Eq. (10), respectively. In panels (c) and (d), the phase shifted ($\phi + 0.2$) blue curves show how such positive and negative kernel values from the drop and the through outputs, respectively. The orange triangle curves show how those values can be amplified by a factor of two using a TIA at the output of the balance photodiode. Those phase shifts are achieved by the application of a current through A_i .

resulting in a modulated intensity equal to A_i :

$$I_{mod} = T_n(\phi_i) |E_0|^2 = A_i, \quad (9)$$

where E_0 is the amplitude of the electric field.

An alternative double waveguide configuration is called the add-drop MRR. The transfer function of the through port light intensity with respect to the input light is:

$$T_p(\phi) = \frac{(ar)^2 - 2r^2a \cos(\phi) + r^2}{1 - 2r^2a \cos(\phi) + (r^2a)^2}; \quad (10)$$

and the transfer function of the drop port light intensity with respect to the input light is:

$$T_d(\phi) = \frac{(1 - r^2)^2 a}{1 - 2r^2a \cos(\phi) + (r^2a)^2}. \quad (11)$$

In the case where the coupling losses are negligible, $a \approx 1$, the relationship between the add-drop through and drop transfer functions is $T_p = 1 - T_d$. In addition, if we connect the through and drop ports into a balanced photodiode and a TIA as in Fig. 3(a), we get an effective transfer function of $g(T_d - T_p)$ where g is the gain of the TIA. Therefore, we get a modulation of:

$$I_{mod} = g(T_d(\phi_i) - T_p(\phi_i)) |E_0|^2 = A_i. \quad (12)$$

At the output of the balanced photodiode, the transfer function of $T_d - T_p$ is shown by the blue triangle curve in Fig. 3(b). The orange circle and green curves are the Lorentzian line shapes centered in the initial phase where MRR is in resonance with the incoming light, described by Eq. (11) and Eq. (10), respectively. Fig. 3(c) and (d) are centered in a modified phase ($\phi + 0.2$), according to a specific value of the current A_i . As we aim to demonstrate how to represent any positive and negative kernel values in analog photonics, we at first incorporate a balanced-PD at the output of the add-drop MRR. Next, we add a TIA and show an example of how values can be amplified by a factor of two

using a TIA at the output of the photodiodes. This architecture can therefore be used to map any kernel value to photonics. In panels (c) and (d), the blue curves show such positive and negative kernel values from the drop and the through outputs, respectively. The orange triangle curves show the TIA transfer function $g(T_d - T_p)$, where g amplifies $T_d - T_p$ by a factor of two.

C. Dot Products With Photonics

The fundamental operation of a convolution is the dot product of two vectorized matrices. Therefore, one needs to understand how to compute a vector dot product using photonics before proposing an architecture capable of performing convolutions.

A wavelength-multiplexed signal consists of k electromagnetic waves, each with angular frequency ω_i , $i = 1, \dots, k$. If it is assumed that each wave has an amplitude of E_0 , a power enveloping function μ_i whose modulation frequency is significantly smaller than ω_i , then the slowly varying envelope approximation and a short-time Fourier transform can be used to derive an expression for the multiplexed signal in the frequency domain:

$$E_{mux}(\omega) = \sum_{i=1}^k E_0 \sqrt{\mu_i} \delta(\omega - \omega_i), \quad (13)$$

where $\delta(\omega - \omega_i)$ is the Dirac delta function and $\mu_i \geq 0$, since power envelopes are not negative. If the enveloping function is prevented from amplifying the electric field, μ_i can further be restricted to the domain $0 \leq \mu_i \leq 1$. Next, we introduce tunable linear filters $H^+(\omega)$ and $H^-(\omega)$ such that when they interact with multiple fields, the following weighted signals are created:

$$\begin{aligned} E_w^-(\omega) &= H^-(\omega) E_{mux}(\omega), \\ E_w^+(\omega) &= H^+(\omega) E_{mux}(\omega). \end{aligned} \quad (14)$$

Assuming that the two signals are fed into a balanced photodiode (balanced PD) with spectral response $R(\omega)$, the induced photocurrent is described by:

$$\begin{aligned} i_{PD} &= \int_{-\infty}^{\infty} d\omega R(\omega) \left(|E_w^+(\omega)|^2 - |E_w^-(\omega)|^2 \right), \\ &= \int_{-\infty}^{\infty} d\omega R(\omega) \left(|H^+(\omega)|^2 - |H^-(\omega)|^2 \right) |E_{mux}(\omega)|^2, \\ &= \sum_{i=0}^{k-1} R(\omega_i) \left(|H^+(\omega_i)|^2 - |H^-(\omega_i)|^2 \right) |E_0|^2 \end{aligned} \quad (15)$$

Assuming that $R(\omega)$ is roughly constant in the area of spectral interest, one can set $A_i = |E_0|^2 R_0 \mu_i$ and $F_i^* = |H^+(\omega_i)|^2 - |H^-(\omega_i)|^2$ resulting in a photocurrent equal to

$$i_{PD} = \sum_{i=1}^k A_i F_i^* = \vec{A} \cdot \vec{F}^*. \quad (16)$$

The through and drop ports of a MRR can be used to implement the linear filters H^+ and H^- such that $|H^+|^2 = T_d$ and $|H^-|^2 = T_p$. Knowing that $T_p = 1 - T_d$ with minimal losses,

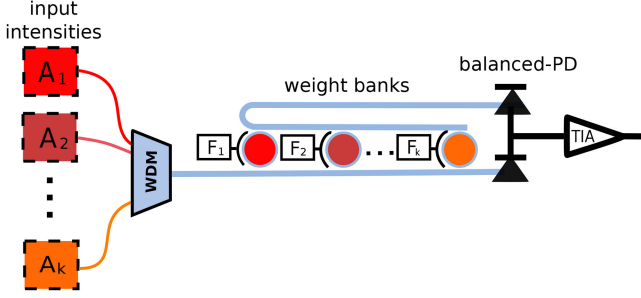


Fig. 4. An electro-optic architecture that performs dot products. A_i ($i = 1, \dots, k$) are input elements encoded in intensities, multiplexed by a WDM and linked to the weight banks via a silicon waveguide. F_i are filter values that modulate the MRRs in the photonic weight bank. Drop and through output ports are connected to a balanced PD, where the matrix multiplication is performed, followed by an amplifier TIA.

we can set a particular weight using:

$$F_i^* = 2T_d(\phi_i) - 1, \quad (17)$$

where the phase, ϕ_i can be obtained by using Eq. (10) and Eq. (11) to get:

$$\phi_i = \arccos \left[-\frac{1}{2r^2a} \left(\frac{2(1-r)^2a}{F_i^* + 1} - 1 - (r^2a)^2 \right) \right], \quad (18)$$

we can see that F_i^* can be between -1 and $+1$. Since T_d is a filter that only represents values between 0 and 1 . In order to perform a dot product with a weight vector \vec{w} whose components are not limited to the range -1 to $+1$, a gain g_{TIA} can be applied to the photocurrent such that:

$$\begin{aligned} \vec{A} \cdot \vec{F} &= g_{TIA} \vec{A} \cdot \vec{F}^* \\ &= g_{TIA} \sum_{i=1}^k A_i F_i^*, \end{aligned} \quad (19)$$

if:

$$g_{TIA} = \max_{1 \leq i \leq k} |F_i|, \quad (20)$$

then,

$$\vec{F} = g_{TIA} \vec{F}^*; \quad (21)$$

assuming that each ϕ_i corresponds to a weighting of w_i^* . This electronic gain can be performed using a TIA, which can be manufactured in a standard CMOS process [28] and packaged or integrated with the photonic chip [23]. A diagram of the electro-optic architecture described in this section is presented in Fig. 4. From now on, this amalgamation of electronic and optical components is referred as a photonic weight bank (PWB). PWBs similar to the one in Fig. 4 have been successfully implemented in the past [11], [29], [30].

We can represent negative inputs between -1 and $+1$ by modifying the power enveloping function to $\mu_i = \frac{1}{2}(x_i + 1)$. If the same set of derivations is followed, we can modify Eq. (21) to be:

$$\vec{x} \cdot \vec{w} = g \left(\sum_{i=1}^k A_i F_i^* + \sum_{i=1}^k E_0 R_0 F_i^* \right). \quad (22)$$

The second term in this sum is a predictable bias current term that can be subtracted before fed into the TIA. This is a disadvantage of supporting negative inputs, as additional optical or electronic control circuitry would need to be designed. Another trade-off is a loss in precision due to a larger range of inputs needing to be represented, analogous to the loss in precision with signed integers for classical computing.

III. PERFORMING CONVOLUTIONS USING PHOTONICS

In this section we introduce a photonic architecture capable of performing convolutions for CNNs. This new architecture is called DEAP. For a maximum number of input channels D_m and a maximum kernel edge length R_m as bounding parameters for DEAP, we represent the range of convolutional parameters that a particular implementation of DEAP can support. If a convolutional parameter described in Table I does not have a complementary bounding parameter, it means that the DEAP architecture can support for arbitrary values of said convolutional parameter.

A. Producing a Single Convolved Pixel

First, we consider an architecture that can produce one convolved pixel at a time. To handle convolutions for kernels with dimensionality up to $R_m \times R_m \times D_m$, we will require R_m^2 lasers with unique wavelengths since a particular convolved pixel can be represented as the dot product of two $1 \times R_m^2$ vectors. To represent the values of each pixel, we require $D_m R_m^2$ modulators (one per kernel value) where each modulator keeps the intensity of the corresponding carrier wave proportional to the normalized input pixel value. The R_m^2 lasers are multiplexed together using WDM, which is then split into D_m separate lines. On every line, there are R_m^2 all-pass MRRs, resulting in $D_m R_m^2$ MRRs in total. Each WDM line will modulate the signals corresponding to a subset of R_m^2 pixels on channel k , meaning that the modulated wavelengths on a particular line correspond to the pixel inputs $(A_{m,n,k})_{n \in [j, j+R_m^2]}^{m \in [i, i+R_m^2]}$ where $k \in [1, D_m]$.

The D_m WDM lines will then be fed into an array of D_m PWBs. Each PWB will contain R_m^2 MRRs with the weights corresponding to the kernel values at a particular channel. An optimized MRR design could support 108 WDM channels. This value depends on both the finesse of the resonator and the channel spacing in linewidth-normalized units [31]. The finesse of our MRRs was estimated to be 368, and the minimum channel spacing is 3.41 linewidths. Therefore $R_m^2 = 108$. For example, the PWB on line k should contain the vectorized weights for the kernel $(F_{m,n,k})_{n \in [1, R_m^2]}^{m \in [1, R_m^2]}$. Each MRR within a PWB should be tuned to a unique wavelength within the multiplexed signal. The outputs of the weight bank array are electrical signals, each proportional to the dot product $(F_{m,n,k})_{n \in [1, R_m^2]}^{m \in [1, R_m^2]} \cdot (A_{p,q,k})_{q \in [j, j+R_m^2]}^{p \in [i, i+R_m^2]}$. Finally, the signals from the weight banks need to be added together. This can be achieved using a passive voltage adder. The output from this adder will therefore be the value of a single convolved pixel. Fig. 5 shows a complete picture of what such an architecture would look like.

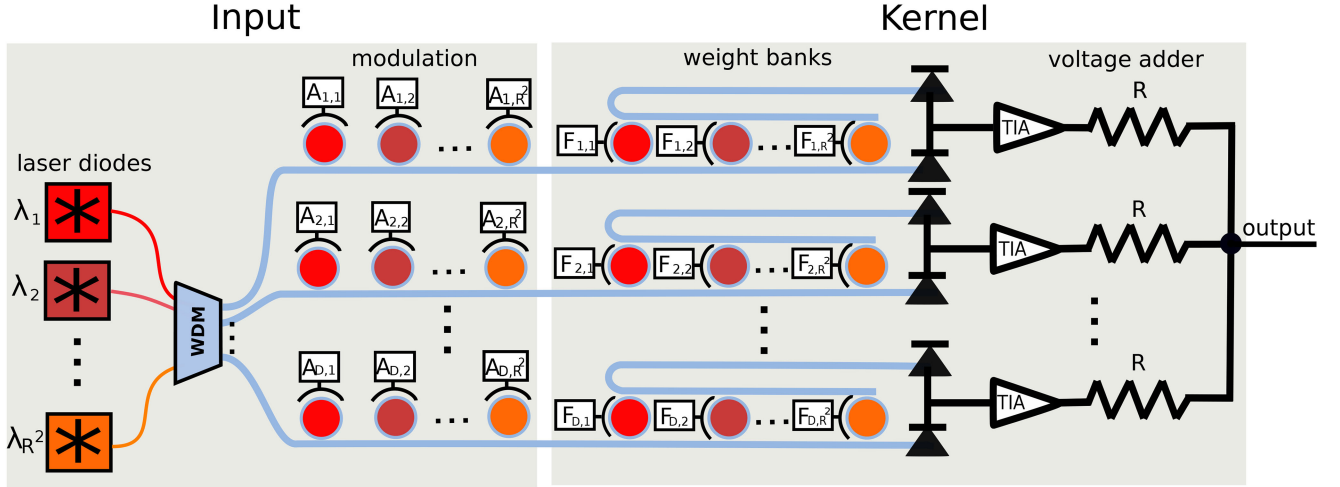


Fig. 5. Photonic architecture for producing a single convolved pixel. Input images are encoded in intensities $A_{l,h}$, where the pixel inputs $A_{m,n,k}$ with $m \in [i, i + R_m]$, $n \in [j, j + R_m]$, $k \in [1, D_m]$ are represented as $A_{l,h}$, $l = 1, \dots, D$ and $h = 1, \dots, R^2$. Considering the boundary parameters, we set $D = D_m$ and $R = R_m$. Likewise, the filter values $F_{m,n,k}$ are represented as $F_{l,h}$ under the same conditions. We use an array of R^2 lasers with different wavelengths λ_h to feed the MRRs. The input and kernel values, $A_{l,h}$ and $F_{l,h}$ modulate the MRRs via electrical currents proportional to those values. Then, the photonic weight banks will perform the dot products on these modulated signals in parallel. Finally, the voltage adder with resistance R adds all signals from the weight banks, resulting in the convolved feature.

To perform a convolution with a kernel edge length less than R_m , one can set $(F_{m,n,k})_{n \in [R+1, R_m]}$ to zero. Similarly, if the dimensionality of the kernel is less than D_m , then the modulators $(A_{m,n,k})_{n \in [1, W]}$ should also be set to zero, with $k \in [D + 1, D_m]$ in this case.

B. Performing a Full Convolution

In the previous section, we have discussed how DEAP can produce a single convolved pixel. In order to perform a convolution of arbitrary size, one would need to stride along the input image and readjust the modulation array. Since the same kernel is applied across the set of inputs, the weight banks do not need to be modified until a new kernel is applied. Fig. 6(a) demonstrates this process on an input with $S = 1$. To handle $S \geq 1$, the inputs being passed in to DEAP should also be strode accordingly. In this approach, the inputs should have been zero padded before being passed into DEAP. In pseudocode, performing a convolution with K filters can be implemented as shown in Algorithm 1.

The DEAP architecture also allows for parallelization by treating the photonic architecture proposed in the previous section as a single output “convolutional unit”. However, by creating n_{conv} instances of these convolutional units, you could produce n_{conv} pixels per cycle by passing in the next set of inputs per unit. This is demonstrated in Fig. 6(b) for $n_{conv} = 2$. The computation of output pixels can be distributed across each convolutional unit, resulting in a runtime complexity of $O(\frac{KHW}{S^2 n_{conv}})$.

IV. PHOTONIC CONVOLUTIONAL NEURAL NETWORKS

In this section, we show how DEAP can be used to run a CNN. CNNs are a type of neural network that were developed

Algorithm 1: Convolutions for CNNs Using DEAP.

```

1:  $A$  is the input image
2:  $F$  is the kernel
3:  $R$  is the edge length of the kernel
4:  $O$  is a memory block to store the convolution
5:  $S$  is the stride
6:  $H$  and  $W$  are the height and width of the input image
7: function CONVOLVE ( $A, F, R, O, S, H, W$ )
8:   for ( $k = 1; k \leq K; k = k + 1$ ) do
9:     load kernel weights from  $F[:, :, :, k]$ 
10:    for ( $h = 1; h \leq H - R + 1; h = h + S$ ) do
11:      for ( $w = 1; w \leq W - R + 1; w = w + S$ ) do
12:        load inputs from  $A[h:\min(h+T,H), w:\min(w+R,W), :]$ 
13:        perform convolution
14:        store results in  $O[h/S, w/S, k]$ 
15:      end for
16:    end for
17:  end for
18: end function

```

for image recognition tasks. A CNN consists of some combination of convolutional, nonlinear, pooling and fully connected layers [32], see Fig. 7(a). As introduced previously, convolutions perform a highly efficient and parallel matrix multiplication using kernels [3]. Furthermore, since kernels are typically smaller than the input images, the feature extraction operation allows efficient edge detection, therefore reducing the amount of memory required to store those features.

CNNs are networks suitable to be implemented in photonic hardware since they demand fewer resources to do matrix

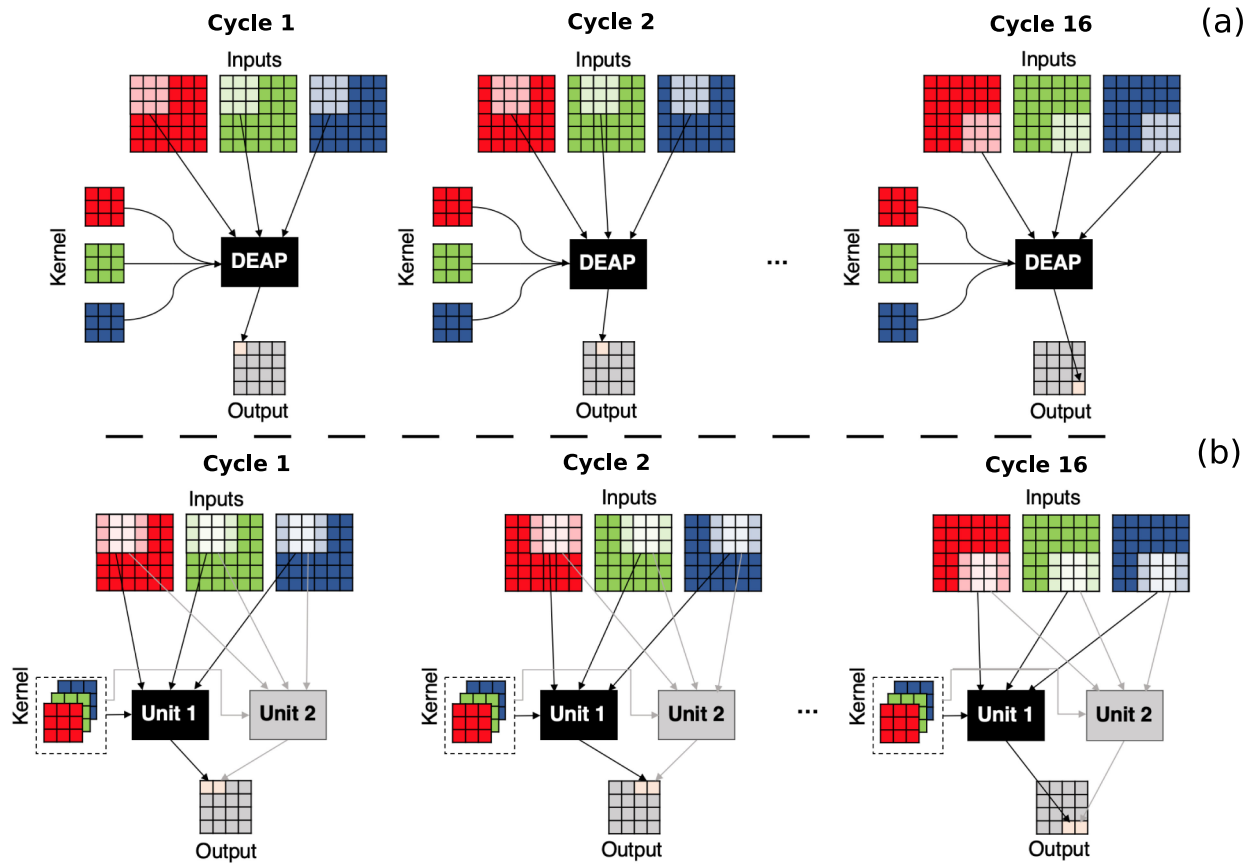


Fig. 6. (a) Cycling through a convolution using DEAP. (b) Performing a convolution with two convolutional units.

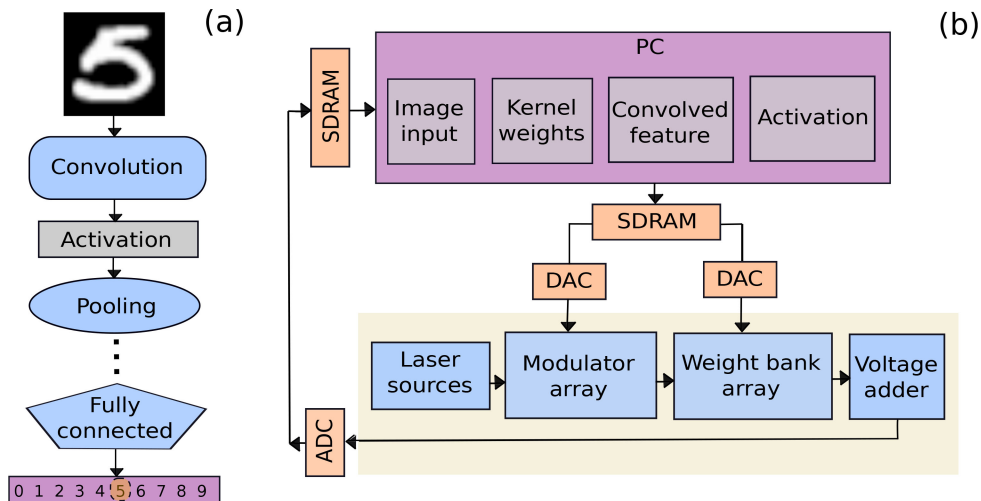


Fig. 7. Block diagrams that describe: (a) A typical CNN, which contains convolutions, activation functions, pooling, and fully connected layers. In this case we exemplify such diagram using MNIST-based recognition task that predicts the number 5; and (b) the DEAP architecture. In the computer (PC) the input image, kernel weights, and convolved features are stored. Also, the commands to implement the activation function off-chip are stored in the PC. The input image, kernel weights, and convolved features are transferred to the chip via DACs from the SDRAM. Then, the convolution is performed on-chip. Finally the output is digitalized via an ADC and stored in a SDRAM connected to the computer.

multiplication and memory usage. The linear operation performed by convolutions allows single feature extraction per kernel. Hence, many kernels are required to extract as many features as possible. For this reason, kernels are usually applied in blocks, allowing the network to extract many different features all at once and in parallel.

In feed-forward networks, it is typical to use a rectified linear unit (ReLU) activation function [33]. Since ReLUs are linear piecewise functions that model an overall nonlinearity, they allow CNNs to be easily optimized during training. The pooling layer introduces a stage where a set of neighbor pixels are encompassed in a single operation. Typically, such operation consists in the application of a function that determines the maximum value among neighboring values. An average operation can be implemented likewise. Both approaches describe max and average pools, respectively. This statistical operation allows for a direct down-sampling of the image, since the dimensions of the object are reduced by a factor of two. From this step, we aim to make our network invariant and robust to small translations of the detected features.

The triplet, convolution-activation-pooling, is usually repeated several times for different kernels, keeping invariant the pooling and activation functions. Once all possible features are detected, the addition of a fully connected layer is required for the classification stage. This layer prepares and shows the solutions of the task.

CNNs are trained by changing the values of the kernels, analogous to how feed-forward neural networks are trained by changing the weighted connections [34]. The estimated kernel and weight values are required in the testing stage. In this work, this stage is performed by our on-chip DEAP CNN. Fig. 7(b) shows a high-level overview of the proposed testing on-chip architecture. Here, the testing input values stored in the PC modulate the intensities of a group of lasers with identical powers but unique wavelengths. These modulated inputs would be sent into an array of photonic weight banks, which would then perform the convolution for each channel. The kernels obtained in the training step are used to modulate these weight banks. Finally, the outputs of the weight banks would be summed using a voltage adder, which produces the convolved feature. This simulator works using the transfer function of the MRRs, through port and drop port summing equations at the balanced PDs, and the TIA gain term to simulate a convolution. The simulator assumes that MRRs transfer functions design is based on the averaged transfer function behavior validated experimentally in prior work [30]. The control accuracy of the MRRs is 6-bits as that has been empirically observed [35]. The MRR self-coupling coefficient is equal to the loss, $r = a = 0.99$ [36] in Eq. (6), Eq. (10) and Eq. (11).

The interfacing of optical components with electronics would be facilitated by the use of digital-to-analog converters (DACs) and analog-to-digital converters (ADCs). The storage of output and retrieving of inputs would be achieved by using memories GDDR SDRAM. The SDRAM is connected to a computer, where the information is already in a digital representation. Then, the implementation of the ReLU nonlinearity and the reuse of the convolved feature to perform the next convolution can be

performed. The idea is to use the same architecture to implement the triplet convolution-activation-pooling on hardware.

In this work, we trained the CNN to perform image recognition on the MNIST dataset. The training stage uses the ADAM optimizer and back-propagation algorithm to compute the gradient function [3]. The optimized parameters to solve MNIST can be categorized in two groups: (i) two $5 \times 5 \times 8$ different kernels and (ii) two fully connected layers of dimensions 800×1 and 10×1 ; and their respective bias terms. These kernels are then defined by eight 5×5 different filters. In the following we use our DEAP CNN simulator to recognize new input images, obtained from a set of 500 images, which are intended to be used for the test step. The process of feature extraction performed by the DEAP CNN is illustrated in Fig. 8(a). As it can be seen in the illustration, a 28×28 input image from the test dataset is filtered by a first $5 \times 5 \times 8$ kernel, using stride one. The output of this process is a $24 \times 24 \times 8$ convolved feature, with a ReLU activation function already applied. Following the same process, the second group of filters is applied to the convolved feature to generate the second output, i.e. a $20 \times 20 \times 8$ convolved feature.

After the second ReLU is applied to the output, average pooling is utilized for invariance and down-sampling of the convolved features. The average pooling is implemented by a 2×2 kernel whose elements are all $1/4$. However, the stride one was kept; therefore the pooled feature has dimensionality $19 \times 19 \times 8$. The down-sampling is implemented offline: from the $19 \times 19 \times 8$ output, a simple algorithm extracts the elements that have even indexes. The result of this process is a $10 \times 10 \times 8$ pooled output. Finally, the first fully connected layer is fed through by the flattened version of the pooled object. The resultant vector feeds the last fully connected layer, where the result of the MNIST classification appears.

The results of the MNIST task solved by our simulated DEAP CNN is shown by Fig. 8(b). For a test set of 500 images, we obtained an overall accuracy of 97.6%. This result was found to be 1% better respect to on-chip Mach-Zehnder interferometers based CNNs [16]. The DEAP-CNN performance was also compared to the results obtained using a standard two-layers CNN. The standard deep network is a 32-bit floating point based CNN that has similar structure to our DEAP-CNN, i.e. two convolutional layers, two ReLU activation functions, one pooling layer and two fully connected layers. This network achieves an overall accuracy of 98.6%. Therefore, we can conclude that our simulator is sufficiently robust despite the 6-bits of precision considered in the DEAP-CNN simulation. In fact, it has been demonstrated that MNIST can be solved using low precision CNNs. For instance, 8-bit floating point based CNNs can solve MNIST with around 97% of accuracy [37], as well as other more complex image and speech classification tasks such as CIFAR10 and BN50-DNN [38].

Noise accumulation can become prominent as analog networks scale in size. Our simulator only incorporates the transfer functions of the individual components and does not simulate noise from analog components; however, our simulator does simulate distortion. Recent work by Ferreira de Lima *et al.* [39] studies the O/E/O links in modulator-class photonic neurons, examining the sources of noise, how it propagates, and how it can

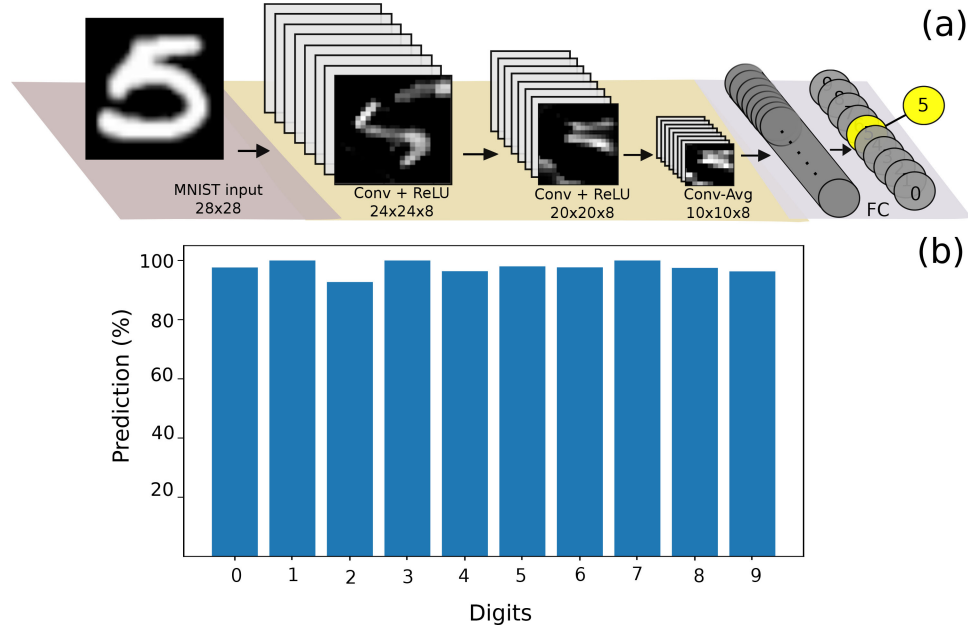


Fig. 8. (a) An illustrative block diagram of the two-layers DEAP CNN solving MNIST. (b) Results of the MNIST task using a simulated DEAP CNN with an overall accuracy of 97.6%.

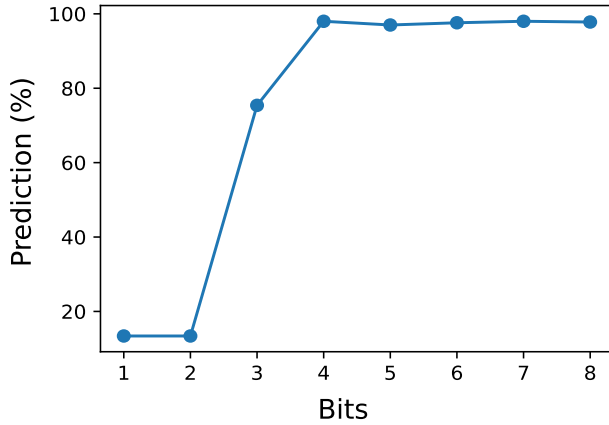


Fig. 9. Performance of the DEAP CNN on MNIST vs. bits of precision.

be suppressed. It finds that the nonlinear transfer function of the modulator can suppress the noise generated by the analog stages in the optoelectronic hardware. This theoretical model could be incorporated in our simulator enabling hardware-constrained learning.

To understand the behavior of our simulator in the presence of distortions, we estimate the overall accuracy of the network when the simulator precision is changed. Fig. 9 shows the impact of the bits number in the simulator on the network's capabilities to recognize new input data. This analysis reveals that out 6-bits DEAP-CNN can admit bit-like distortions of 2-bits without performance flaws. In general, performance for DEAP-CNNs designed with equal or less than 4-bits is significantly diminished. Hence, the challenges brought in by bit fluctuations could be surpassed by our proposed 6-bits architecture.

V. ENERGY AND SPEED ANALYSES

In the following, we perform an energy-speed benchmark test, where we compare the performance of DEAP with actual GPU runtimes from DeepBench benchmarks.

A. Energy and Speed Estimation

The energy used by a single DEAP convolutional unit depends on the R and D parameters. As introduced above in Sec. 3.A, the 108-wavelength limitation for MRRs constrains the maximum R to be 10, as each multiplexed waveguide will store R^2 signals. The number of MRRs used in the modulator array is equal to R^2D , meaning that only certain D and R^2 values are allowed for a finite number of MRRs. Assuming that a maximum of 1024 MRRs can be manufactured in the modulator array, a convolutional unit can support a large kernel size with a limited number of channels, $R = 10$, $D = 12$, or a small kernel size with a large number of channels, $R = 3$, $D = 113$. We will consider both edge cases to get a range of energy consumption values. For the smaller convolution size, we will have R^2 lasers, R^2 MRRs and DACs in the modulator array, R^2D MRRs and D TIAs in the weight bank array and one ADC to convert back into digital signal. With 100 mW per laser, 19.5 mW per MRR, 26 mW per DAC, 17 mW per TIA [40] and 76 mW per ADC, we get an energy usage of 112 W for the large kernel size and 95 W for the smaller kernel size. Therefore, we estimate a single convolution unit to use around 100 W when 1024 modulators are used to represent inputs.

The time it takes for light to propagate from the WDM to before the balanced PDs is estimated by the following equation:

$$t_{prop} = \frac{k \cdot 2\pi r_{MRR} \cdot n}{c} \quad (23)$$

TABLE II
BENCHMARKING PARAMETERS FOR DEAP

W	H	D	N	K	R_w	R_h	S
700	161	1	4	32	5	20	2
112	112	64	8	128	3	3	1
7	7	832	16	256	1	1	1

TABLE III
BENCHMARKED GPUS WITH POWER CONSUMPTION

GPU	Power Usage (W)
AMD Vega FE [46]	375
AMD MI25 [47]	300
NVIDIA Tesla P100 [48]	250
NVIDIA GTX 1080 Ti [49]	250

where c is the speed of light, n is the index of refraction of the waveguide, $2\pi r_{MRR}$ is the circumference of the MRR, and k is the number of MRRs. Assuming 100 MRRs with a radius of radius of $10 \mu\text{m}$ [11], [41] and a silicon index of refraction of 3.4, the PWB gets a propagation time of around 71 ps and a throughput of $1/t_{prop} = 14 \text{ GS/s}$. The balance PD and the TIA have a throughput of 25 GS/s [40] and 10 GS/s [42], respectively. The DACs [43] and ADCs [44] both operate at 5 GS/s and support 7-bits. The GDDR6 SDRAM operates at 16 GS/s with a 256-bit bus size [45]. In addition, an individual MRR can be modulated at speeds of 128 GS/s [41]. Consequently, the speed of the system is limited by the throughput of the DACs/ADCs, resulting in DEAP producing a single convolved pixel at 5 GS/s or $t = 200 \text{ ps}$.

B. DEAP Performance

In order to benchmark the speed of DEAP, we use DeepBench [19] as a base for the analysis. DeepBench is an empirical dataset that contains how long various types of GPUs took to perform a convolution for a given set of convolutional parameters. Table II contains the parameters used for each of these benchmarks, and Table III contains the power consumption.

The speeds of various GPUs were directly taken from Ref. [19], while the speed of the convolution was estimated using the following equation:

$$t_{runtime} = 200 \text{ ps} \times \frac{NK}{n_{conv}} \left(\frac{H-R}{S} + 1 \right) \left(\frac{W-R}{S} + 1 \right). \quad (24)$$

In some of the benchmarks, the kernels edge lengths were not equal, hence the parameters R_w and R_h which correspond to the width and height of the kernels. For each of the selected benchmarks, the parameters $R^2 D \leq 1024$, meaning that the convolutional network is compatible with DEAP implementations.

The estimated DEAP runtimes using one and two convolutional units were plotted against actual DeepBench runtimes in Fig. 10. From this, we can see that using two convolutional units performs slightly better than all the GPU benchmarks. While mean GPUs power consumption is 295 W, DEAP with a single convolutional unit uses about 110 W. Therefore, DEAP can perform convolutions between 1.4 and $7.0 \times$ faster than the

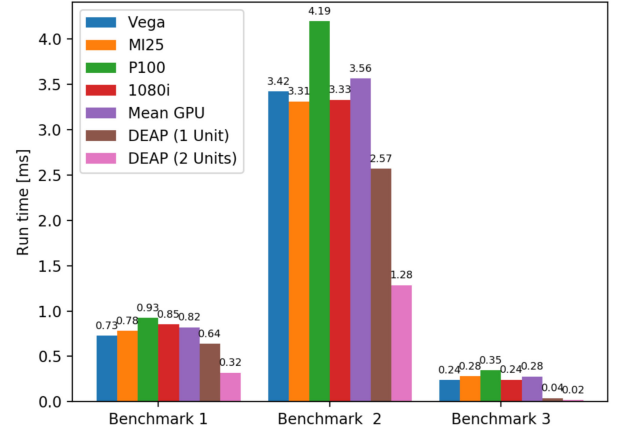


Fig. 10. Estimated DEAP convolutional runtime compared to actual GPU runtimes from DeepBench benchmarks.

mean GPU runtime while using 63% less energy. Using two convolutional units doubles the speed of DEAP, meaning that DEAP can be between 2.8 and $14 \times$ faster than a conventional GPU while using almost 25% less energy. DEAP with a single unit performing at a speed somewhat similar to the GPUs is expected.

VI. CONCLUSION

We have proposed a photonic network, DEAP, suited for convolutional neural networks. DEAP was estimated to perform convolutions between 2.8 and $14 \times$ faster than a GPU while roughly using 25% less energy. A linear increase in processing speeds corresponds to a linear increase in energy consumption, allowing for DEAP to be as scalable as electronics.

High-level software simulations have shown that DEAP is theoretically capable of performing a convolution. We demonstrate that DEAP-CNN engine, which was limited to only storing 64 values between -1 and $+1$, is capable of solving MNIST handwritten recognition task with an overall accuracy of 97.6%. The largest bottlenecks is the I/O interfacing with digital systems via DACs and ADCs. If photonic DACs [50] and ADCs [51] are to be built with higher bit-precisions, the speedup over GPUs could be even higher. If higher bit precision photonic DACs and ADCs are able to be built, replacing the electronic components with optical ones can significantly decrease the runtime.

In order to realize a physical implementation, there are a number of issues that still need to be solved. Packaging a silicon photonic with an electronic chip with high I/O count is a challenging RF engineering task, but it is a central thrust in the roadmap for silicon photonic foundries [23]. There also needs to be control circuitry that routes the outputs of the SDRAM into the relevant DACs and from the ADCs into the SDRAM. Since we assume that the control circuitry can operate significantly faster than a memory access, we believe it will have a negligible impact on the overall throughput. Another issue is that DEAP processes data in the analog domain, whereas GPUs perform floating point arithmetic. Though floating-point arithmetic does have some degree of error due to rounding in the mantissa,

their errors are deterministic and predictable. On the other hand, the errors from photonics are due to stochastic shot, spectral, Johnson-Nyquist and flicker noises, as well as quantization noise in the ADC, and distortion from the RF signals applied to the modulators. However, artificially adding random noise to CNNs have been shown to reduce over-fitting [52], meaning that some degree of stochastic behaviour is tolerable in the domain of machine learning problems.

Finally, MRRs have only been shown to have up to 6-bits of precision, which is significantly smaller than the range precision supported by even half-precision (16-bit) floating point representations. In conclusion, photonics has the potential to perform convolutions at speeds faster than top-of-the-line GPUs while having a lower energy consumption. Moving forward, the greatest challenges to overcome have to do with increasing the precision of photonic components so that they are comparable to classical floating-point representations. Overall, silicon photonics has the potential to outperform conventional electronic hardware for convolutions while having the ability to scale up in the future.

ACKNOWLEDGMENT

The authors thank Prof. Sudip Shekhar, Mohammed Al-Qadasi, Hugh Morison and Matthew Filipovich for edits and suggestions.

REFERENCES

- [1] X. Li, G. Zhang, H. H. Huang, Z. Wang, and W. Zheng, "Performance analysis of GPU-based convolutional neural networks," in *Proc. 45th Int. Conf. Parallel Process.*, Aug. 2016, pp. 67–76.
- [2] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," 2014, *arXiv:1405.3866*.
- [3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [4] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [5] F. Dupont, A. Smerieri, A. Akrou, M. Haelterman, and S. Massar, "Fully analogue photonic reservoir computer," *Scientific Rep.*, vol. 6, 2016, Art. no. 22381. [Online]. Available: <http://dx.doi.org/10.1038/srep22381>
- [6] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, "Parallel photonic information processing at gigabyte per second data rates using transient states," *Nature Commun.*, vol. 4, 2013, Art. no. 1364. [Online]. Available: <http://dx.doi.org/10.1038/ncomms2368>
- [7] K. Vandoorne *et al.*, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature Commun.*, vol. 5, 2014, Art. No. 3541. [Online]. Available: <http://dx.doi.org/10.1038/ncomms4541>
- [8] L. Larger *et al.*, "Photonic information processing beyond turing: An optoelectronic implementation of reservoir computing," *Opt. Express*, vol. 20, no. 3, pp. 3241–3249, Jan. 2012. [Online]. Available: <http://dx.doi.org/10.1364/OE.20.003241>
- [9] P. R. Prucnal and B. J. Shastri, *Neuromorphic Photonics*. Boca Raton, FL, USA: CRC Press, 2017.
- [10] A. N. Tait, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, "Broadcast and weight: An integrated network for scalable photonic spike processing," *J. Lightw. Technol.*, vol. 32, no. 21, pp. 4029–4041, Nov. 2014.
- [11] A. N. Tait *et al.*, "Microring weight banks," *IEEE J. Sel. Topics Quantum Electron.*, vol. 22, no. 6, pp. 312–325, Nov. 2016.
- [12] A. N. Tait *et al.*, "A silicon photonic modulator neuron," *Phys. Rev. Applied*, vol. 11, no. 6, p. 064043, Jun. 2019.
- [13] T. W. Hughes, M. Minkov, Y. Shi, and S. Fan, "Training of photonic neural networks through in situ backpropagation and gradient measurement," *Optica*, vol. 5, no. 7, pp. 864–871, Jul. 2018, doi: [10.1364/OPTICA.5.000864](https://doi.org/10.1364/OPTICA.5.000864).
- [14] Y. Shen *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photon.*, vol. 11, pp. 441–446, Jun. 2017. [Online]. Available: <http://dx.doi.org/10.1038/nphoton.2017.93>
- [15] T. F. De Lima, *et al.*, "Machine learning with neuromorphic photonics," *J. Lightw. Technol.*, vol. 37, no. 5, pp. 1515–1534, Mar. 2019.
- [16] H. Bagherian *et al.*, "On-chip optical convolutional neural networks," 2018, *arXiv:1808.03303*. [Online]. Available: <http://arxiv.org/abs/1808.03303>
- [17] A. Mehrabian, Y. Al-Kabani, V. J. Sorger, and T. A. El-Ghazawi, "PCNNA: A photonic convolutional neural network accelerator," in *Proc. 31st IEEE Int. System-on-Chip Conf. (SOCC)*, Arlington, VA, USA, 2018, pp. 169–173, doi: [10.1109/SOCC.2018.8618542](https://doi.org/10.1109/SOCC.2018.8618542).
- [18] W. Liu *et al.*, "Holylight: A nanophotonic accelerator for deep learning in data centers," in *Proc. Des., Autom. Test Eur. Conf. Exhib.*, Mar. 2019, pp. 1483–1488.
- [19] Baidu Research, "DeepBench." [Online]. Available: <https://github.com/baidu-research/DeepBench>
- [20] V. Bangari, B. Marquez, and B. J. Shastri, "DEAP," 2019. [Online]. Available: <https://github.com/Shastri-Lab/DEAP>
- [21] G. Tan *et al.*, "Fast implementation of DGEMM on Fermi GPU," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, New York, NY, USA: ACM, 2011, pp. 35:1–35:11. [Online]. Available: <http://doi.acm.org/10.1145/2063384.2063431>
- [22] S. Chetlur *et al.*, "cuDNN: Efficient primitives for deep learning," 2014, *arXiv:1410.0759*. [Online]. Available: <http://arxiv.org/abs/1410.0759>
- [23] A. Rahim, T. Spuesens, R. Baets, and W. Bogaerts, "Open-access silicon photonics: Current status and emerging initiatives," *Proc. IEEE*, vol. 106, no. 12, pp. 2313–2330, Dec. 2018.
- [24] M. A. Nahmias, B. J. Shastri, A. N. Tait, T. F. de Lima, and P. R. Prucnal, "Neuromorphic photonics," *Opt. Photon. News*, vol. 29, no. 1, pp. 34–41, Jan. 2018. [Online]. Available: https://www.osapublishing.org/abstract.cfm?uri=CLEO_AT-2019-JM3M.3
- [25] G. T. Reed, G. Mashanovich, F. Y. Gardes, and D. J. Thomson, "Silicon optical modulators," *Nature Photon.*, vol. 4, p. 518, Jul. 2010. [Online]. Available: <https://doi.org/10.1038/nphoton.2010.179>
- [26] W. Bogaerts *et al.*, "Silicon microring resonators," *Laser Photon. Rev.*, vol. 6, no. 1, pp. 47–73, Jan. 2012. [Online]. Available: <http://doi.wiley.com/10.1002/lpor.201100017>
- [27] H. Jayatilaka *et al.*, "Wavelength tuning and stabilization of microring-based filters using silicon in-resonator photoconductive heaters," *Opt. Express*, vol. 23, no. 19, pp. 25084–25097, Sep. 2015. [Online]. Available: <http://www.opticsexpress.org/abstract.cfm?URI=oe-23-19-25084>
- [28] H. Zheng, R. Ma, and Z. Zhu, "A linear and wide dynamic range transimpedance amplifier with adaptive gain control technique," *Analog Integr. Circuits Signal Process.*, vol. 90, no. 1, pp. 217–226, Jan. 2017. [Online]. Available: <https://doi.org/10.1007/s10470-016-0867-1>
- [29] M. Lipson, "Guiding, modulating, and emitting light on silicon-challenges and opportunities," *J. Lightw. Technol.*, vol. 23, no. 12, pp. 4222–4238, Dec. 2005.
- [30] A. N. Tait *et al.*, "Feedback control for microring weight banks," *Opt. Express*, vol. 26, no. 20, pp. 26422–26443, Oct. 2018. [Online]. Available: <http://www.opticsexpress.org/abstract.cfm?URI=oe-26-20-26422>
- [31] A. Tait, "Silicon photonic neural networks," Ph.D. dissertation, Princeton Univ., Princeton, NJ, 2018. [Online]. Available: <http://arks.princeton.edu/ark:/88435/dsp01vh53wz43k>
- [32] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv: 1511.08458*.
- [33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: <http://doi.acm.org/10.1145/3065386>
- [34] K. Mehrotra, C. K. Mohan, and S. Ranka, *Elements of Artificial Neural Networks*. Cambridge, MA, USA: MIT Press, 1997.
- [35] P. Y. Ma *et al.*, "Photonic principal component analysis using an on-chip microring weight bank," *Opt. Express*, vol. 27, no. 13, pp. 18329–18342, Jun. 2019. [Online]. Available: <http://www.opticsexpress.org/abstract.cfm?URI=oe-27-13-18329>
- [36] Y. Tan and D. Dai, "Silicon microring resonators," *J. Opt.*, vol. 20, no. 5, Apr. 2018, Art. no. 054004. [Online]. Available: <https://doi.org/10.1088/2F2040-8986/2Faaba20>
- [37] M. Gallus and A. Nannarelli, "Handwritten digit classification using 8-bit floating point based convolutional neural networks," *Tech. Univ. Denmark, Lyngby, Denmark, DTU Compute Tech. Rep.*-2018, 2018.
- [38] N. Wang, J. Choi, D. Brand, C.-Y. Chen, and K. Gopalakrishnan, "Training deep neural networks with 8-bit floating point numbers," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA: Curran Associates Inc., 2018, pp. 7686–7695. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3327757.3327866>

- [39] T. Ferreira de Lima *et al.*, “Noise analysis of photonic modulator neurons,” Jul. 2019, *arXiv:1907.07325*.
- [40] Z. Huang *et al.*, “25 Gbps low-voltage waveguide Si-Ge avalanche photodiode,” *Optica*, vol. 3, no. 8, pp. 793–798, Aug. 2016. [Online]. Available: <http://www.osapublishing.org/optica/abstract.cfm?URI=optica-3-8-793>
- [41] J. Sun *et al.*, “A 128 Gb/s PAM4 silicon microring modulator with integrated thermo-optic resonance tuning,” *J. Lightw. Technol.*, vol. 37, no. 1, pp. 110–115, Jan. 2019.
- [42] M. Atef and H. Zimmermann, “Low-power 10 Gb/s inductorless inverter based common-drain active feedback transimpedance amplifier in 40 nm CMOS,” *Anal. Integr. Circuits Signal Process.*, vol. 76, no. 3, pp. 367–376, Sep. 2013. [Online]. Available: <http://dx.doi.org/10.1007/s10470-013-0117-8>
- [43] B. Sedighi, M. Khafaji, and J. C. Scheytt, “Low-power 8-bit 5-GS/s digital-to-analog converter for multi-gigabit wireless transceivers,” *Int. J. Microw. Wireless Technol.*, vol. 4, no. 3, pp. 275–282, 2012.
- [44] J. Fang *et al.*, “A 5-GS/s 10-b 76-mW time-interleaved SAR ADC in 28 nm CMOS,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 64, no. 7, pp. 1673–1683, Jul. 2017.
- [45] Micron Technology, “GDDR6 SGRAM MT61K256M32: 2 Channels x16/x8 GDDR6 SGRAM.” [Online]. Available: <https://www.micron.com/products/graphics-memory/gddr6/part-catalog/mt61k256m32je-12>.
- [46] Advanced Micro Devices, “Radeon Vega frontier edition (liquid cooled).” [Online]. Available: <https://www.amd.com/en/products/professional-graphics/radeon-vega-frontier-edition-liquid-cooled>.
- [47] Advanced Micro Devices, “Radeon instinct MI25 accelerator.” [Online]. Available: <https://www.amd.com/en/products/professional-graphics/instinct-mi25>
- [48] NVIDIA Corporation, “NVIDIA Tesla P100 GPU accelerator.” [Online]. Available: <https://images.nvidia.com/content/tesla/pdf/nvidia-tesla-p100-PCIe-datasheet.pdf>
- [49] NVIDIA Corporation, “Geforce GTX 1080 Ti.” [Online]. Available: <https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080-ti/>
- [50] F. Zhang, B. Gao, X. Ge, and S. Pan, “Simplified 2-bit photonic digital-to-analog conversion unit based on polarization multiplexing,” *Opt. Eng.*, vol. 55, 2015, Art. no. 031115.
- [51] M. A. Piqueras, P. Villalba, J. Puche, and J. Martí, “High performance photonic adc for space and defence applications,” in *Proc. IEEE Int. Conf. Microw., Commun., Antennas Electron. Syst.*, Nov. 2011, pp. 1–6.
- [52] Z. You, J. Ye, K. Li, and P. Wang, “Adversarial noise layer: Regularize neural network by adding noise,” 2018, *arXiv:1805.08000*. [Online]. Available: <http://arxiv.org/abs/1805.08000>

Viraj Bangari is currently working towards the B.A.Sc. degree in engineering physics with a computing specialization with Queen’s University, ON, Canada. He has an experience working with Azure Compute Team at Microsoft and with Core Build Technologies Team at Apple. His current research interests are in the fields of photonics, distributed and parallel computing, programming languages and compilers, operating systems, and novel computing hardware.

Bicky A. Marquez (M’19) received the Ph.D. degree in optics/photonics from Bourgogne-Franche-Comté University, France, in 2018. She also holds the bachelor’s (2012) and master’s (2014) degrees from the Central University of Venezuela (UCV) and the Venezuelan Institute for Scientific Research (IVIC), respectively. Her research interests are focused on nonlinear and complex dynamical systems, machine learning, and AI photonic hardware. She has authored and co-authored eight refereed papers and presented research at ten conferences. Dr. Marquez is the recipient of the 2019 Queen’s Postdoctoral Fund, Franche-Comté Recherche d’Excellence (Excellence in Research) scholarship (2014–2017), UCV Student Merit Award in Research (2012), IVIC Excellence scholarship (2012–2014), and ONCTI Incentive Program for Research and Innovation: Researcher A-I (2013).

Heidi Miller received the B.S.E. degree in chemical and biological engineering from Princeton University in 2018.

Alexander N. Tait received the Ph.D. degree from the Lightwave Communications Research Laboratory, Department of Electrical Engineering, Princeton University, Princeton, NJ, USA, advised by Professor Paul Prucnal. He also received the B.Sci.Eng. (Honors) degree in electrical engineering from Princeton University in 2012. His research interests include silicon photonics, optical signal processing, optical networks, and neuromorphic engineering.

Dr. Tait is a recipient of the National Science Foundation Graduate Research Fellowship and is a Student Member of the IEEE Photonics Society and the Optical Society of America (OSA). He is the recipient of the Award for Excellence from the Princeton School of Engineering and Applied Science (SEAS), the Optical Engineering Award of Excellence from the Princeton Department of Electrical Engineering, the Best Student Paper Award at the 2016 IEEE Summer Topicals Meeting Series, and the Class of 1883 Writing Prize from the Princeton Department of English. He has authored nine refereed papers and a book chapter, presented research at 13 technical conferences, and contributed to the textbook *Neuromorphic Photonics*.

Mitchell A. Nahmias received the B. S. (Honors) degree in electrical engineering with a Certificate in Engineering Physics in 2012 and the M.A. degree in electrical engineering in 2014, both from Princeton University. He is currently pursuing the Ph.D. degree as a member of the Princeton Lightwave Communications Laboratory. He was a Research Intern with the MIRTH Center, Princeton, NJ, USA, during the summers of 2011–2012 and with L-3 Photonics during the summer of 2014 in Carlsbad, CA. His research interests include laser excitability, photonic integrated circuits, unconventional computing, and neuromorphic photonics.

Mr. Nahmias is a Student Member of the IEEE Photonics Society and the Optical Society of America (OSA) and has authored or co-authored more than 50 journal or conference papers. He was the recipient of the Best Engineering Physics Independent Work Award (2012), the National Science Foundation Graduate Research Fellowship (NSF GRFP), the Best Paper Award at the IEEE Photonics Conference 2014 (third place), and the Best Paper Award at the 2015 IEEE Photonics Society Summer Topicals Meeting Series (first place). He is also a contributing author of the textbook *Neuromorphic Photonics*.

Thomas Ferreira de Lima received the bachelor’s degree and the Ingénieur Polytechnicien master’s degree from Ecole Polytechnique, Palaiseau, France, with a focus on physics for optics and nanosciences. He is working toward the Ph.D. degree in electrical engineering with the Lightwave Communications Group, Department of Electrical Engineering, Princeton University, Princeton, NJ. His research interests include integrated photonic systems, nonlinear signal processing with photonic devices, spike-timing-based processing, ultrafast cognitive computing, and dynamical light-matter neuro-inspired learning and computing. He has authored or co-authored more than 40 journal or conference papers and contributed to four major open-source projects, and is a contributing author of the textbook *Neuromorphic Photonics*.

Hsuan-Tung Peng received the B.S. degree in physics from National Taiwan University in 2015 and the M.A. degree in electrical engineering from Princeton University in 2018. He is now pursuing the Ph.D. degree with Princeton University, Princeton NJ, USA. His current research interests include neuromorphic photonics, photonic integrated circuits, and optical signal processing.

Paul R. Prucnal (LF'19) received the A.B. in mathematics and physics from Bowdoin College, graduating *summa cum laude*. He then earned the M.S., M.Phil., and Ph. D. degrees in electrical engineering from Columbia University. After his doctorate, he joined the faculty at Columbia University, where, as a member of the Columbia Radiation Laboratory, he performed groundbreaking work in OCDMA and self-routed photonic switching. In 1988, he joined the faculty at Princeton University. His research on optical CDMA initiated a new research field in which since then more than 1000 papers have been published, exploring applications ranging from information security to communication speed and bandwidth. In 1993, he invented the "Terahertz Optical Asymmetric Demultiplexer," the first optical switch capable of processing terabit per second (Tb/s) pulse trains. He has authored or co-authored more than 350 journal articles and book chapters and holds 28 U.S. patents. He is the author of the book *Neuromorphic Photonics* and an editor for the book *Optical Code Division Multiple Access: Fundamentals and Applications*. He was an Area Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS. He is a Fellow of the Optical Society of America (OSA) and the National Academy of Inventors (NAI), and a member of honor societies, including Phi Beta Kappa and Sigma Xi. He was the recipient of the 1990 Rudolf Kingslake Medal for his paper entitled "Self-routing photonic switching with optically-processed control," received the Gold Medal from the Faculty of Mathematics, Physics and Informatics, Comenius University, for leadership in the field of Optics 2006 and has won multiple teaching awards at Princeton, including the E-Council Lifetime Achievement Award for Excellence in Teaching, the School of Engineering and Applied Science Distinguished Teacher Award, and The President's Award for Distinguished Teaching. He has been instrumental in founding the field of neuromorphic photonics and developing the "photonic neuron," a high-speed optical computing device modeled on neural networks as well as integrated optical circuits to improve the wireless signal quality by cancelling radio interferences.

Bhavin J. Shastri (SM'19) is an Assistant Professor of Engineering Physics with Queen's University, Canada. He earned the Honours B.Eng. (with distinction), M.Eng., and Ph.D. degrees in electrical engineering (photonics) from McGill University, Canada, in 2005, 2007, and 2012, respectively. He was an NSERC and Banting Postdoctoral Fellow (2012–2016) and an Associate Research Scholar (2016–2018) with Princeton University. With research interests in silicon photonics, photonic integrated circuits, neuromorphic computing, and machine learning, he has published more than 120 journal and conference publications and 3 book chapters. He is a co-author of the book *Neuromorphic Photonics* (CRC Press, 2017).

Dr. Shastri is the recipient of the 2014 Banting Postdoctoral Fellowship from the Government of Canada, the 2012 D. W. Ambridge Prize for the top graduating Ph.D. student, an IEEE Photonics Society 2011 Graduate Student Fellowship, a 2011 NSERC Postdoctoral Fellowship, a 2011 SPIE Scholarship in Optics and Photonics, a 2008 NSERC Alexander Graham Bell Canada Graduate Scholarship, including the Best Student Paper Awards at the 2014 IEEE Photonics Conference, and 2010 IEEE Midwest Symposium on Circuits and Systems, and the 2004 IEEE Computer Society Lance Stafford Larson Outstanding Student Award and 2003 IEEE Canada Life Member Award.